TOTTOT" TBZE2660

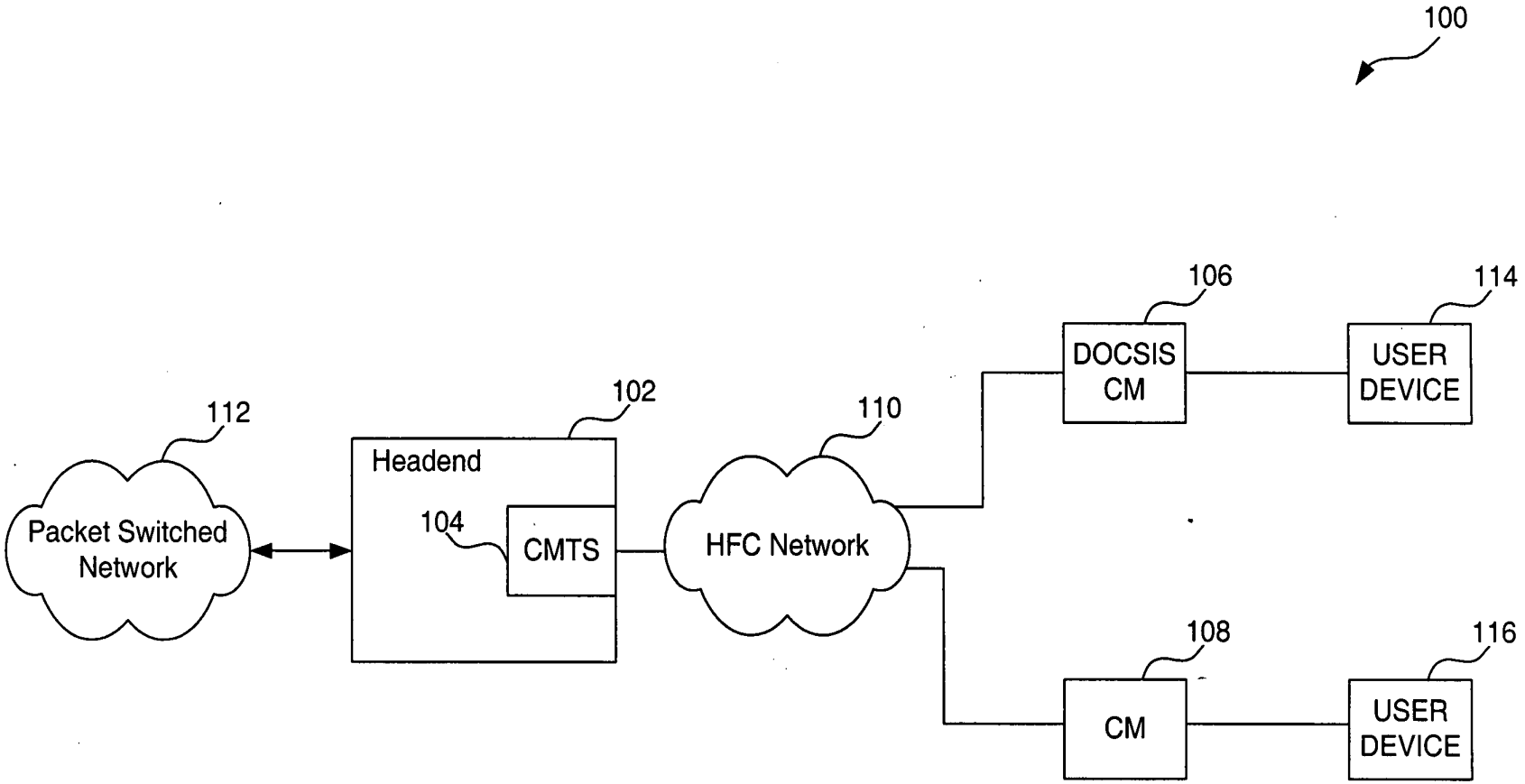
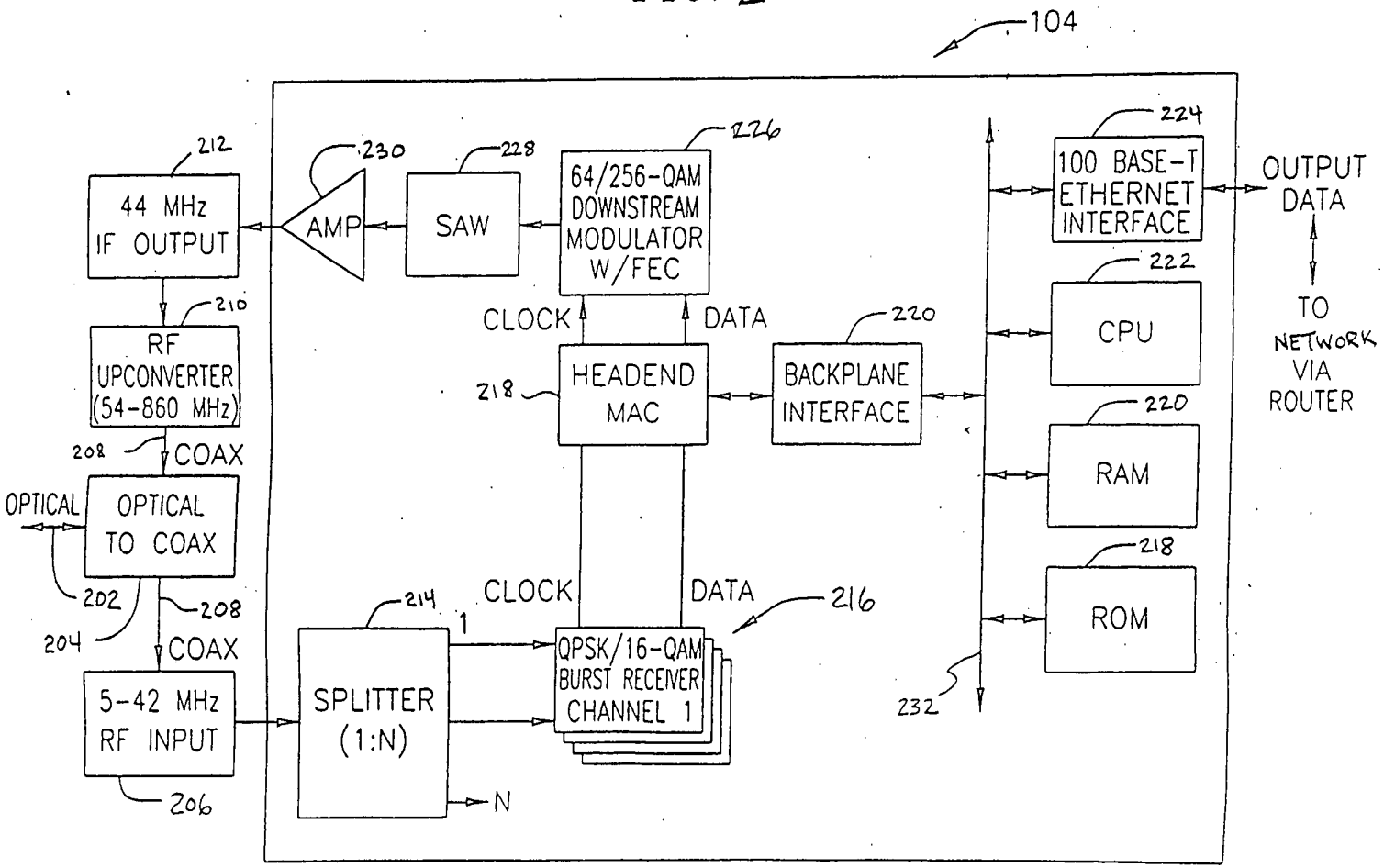


FIG. 1

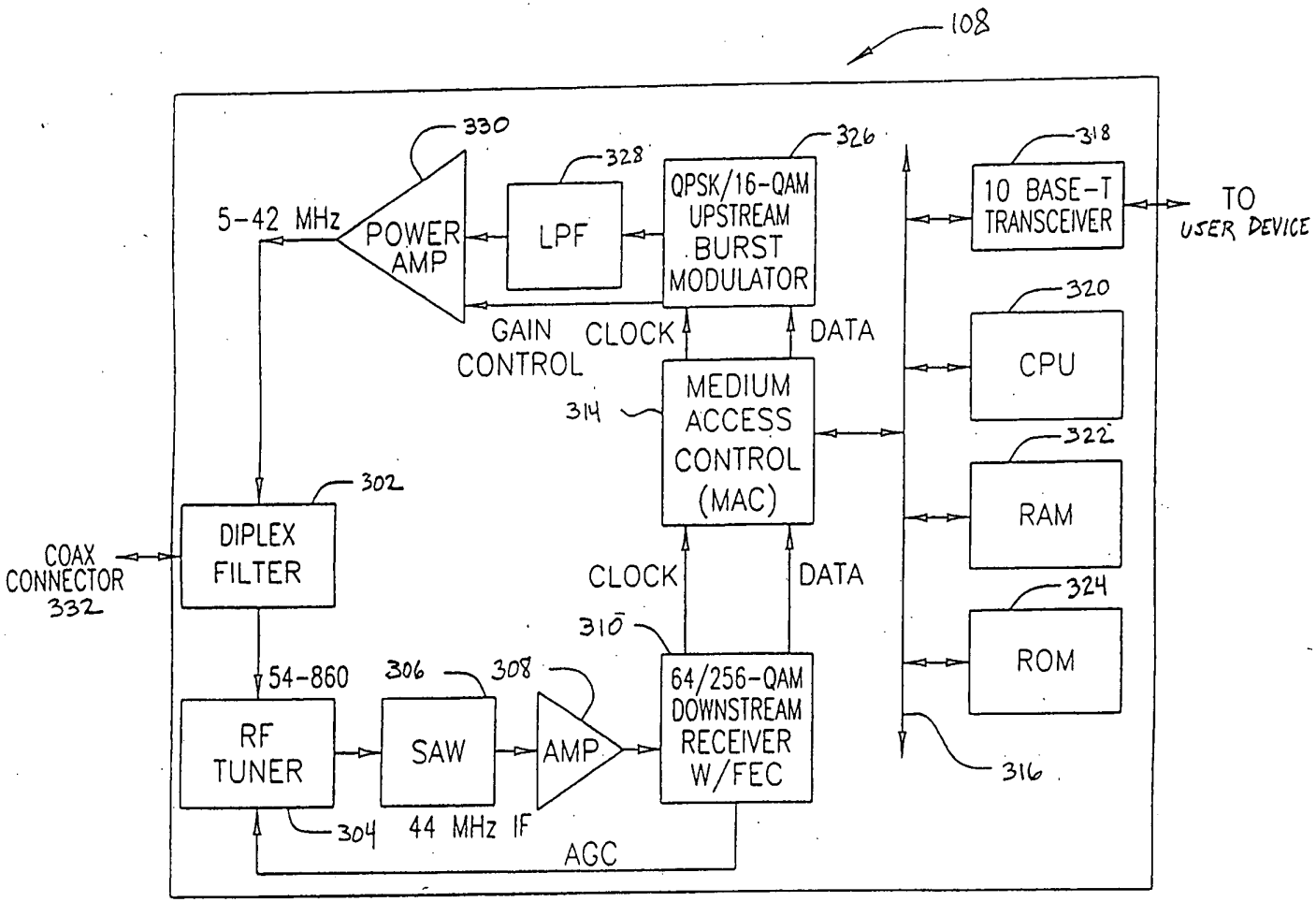
TOTTOT" TBZE2660

FIG. 2



EOFFOF" TBZEZ660

FIG. 3



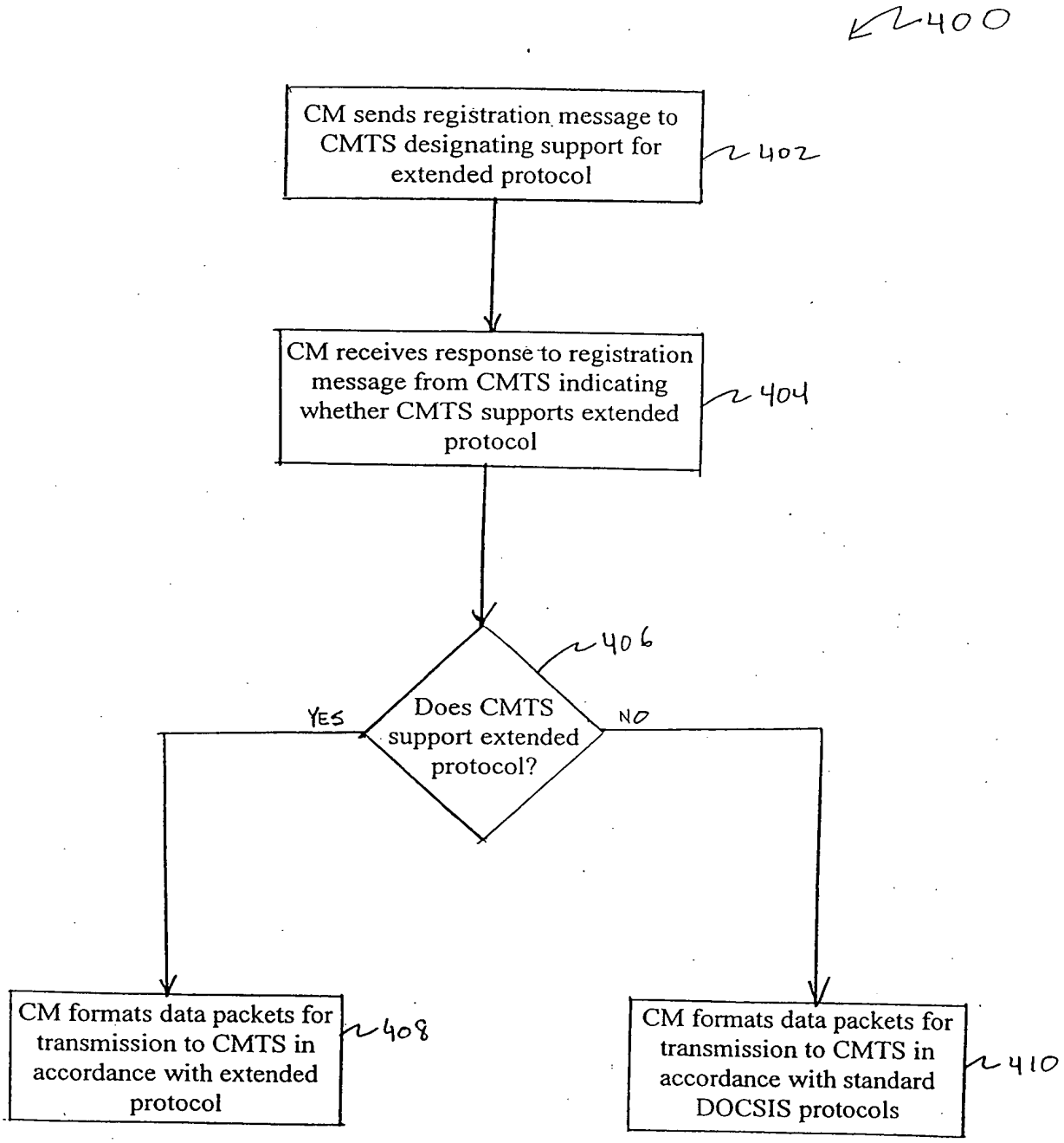

400
CM sends registration message to CMTS designating support for extended protocol
402
CM receives response to registration message from CMTS indicating whether CMTS supports extended protocol
404
406
Does CMTS support extended protocol?
YES
NO
CM formats data packets for transmission to CMTS in accordance with extended protocol
408
CM formats data packets for transmission to CMTS in accordance with standard DOCSIS protocols
410

FIG. 4

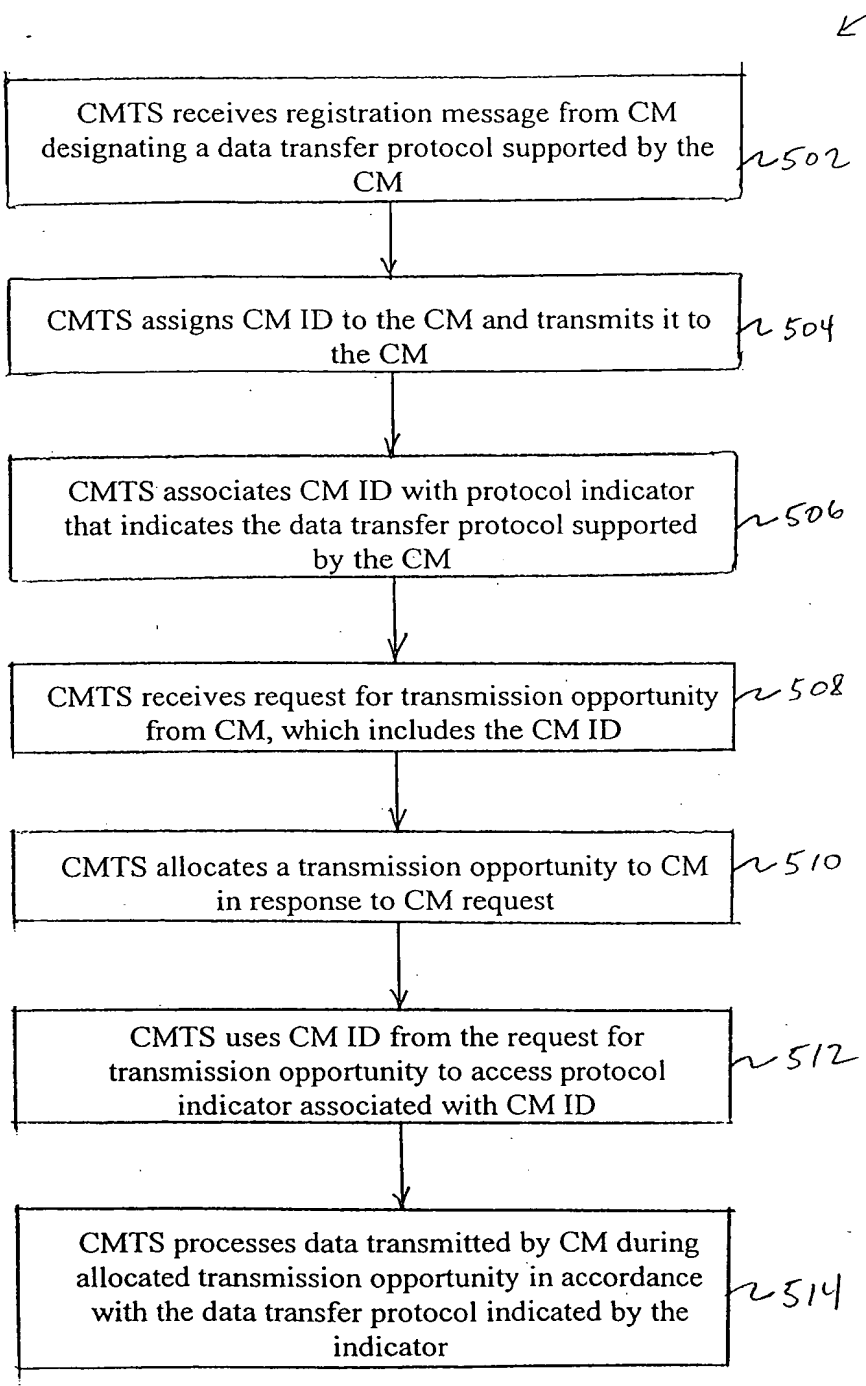

500
CMTS receives registration message from CM designating a data transfer protocol supported by the CM
502
CMTS assigns CM ID to the CM and transmits it to the CM
504
CMTS associates CM ID with protocol indicator that indicates the data transfer protocol supported by the CM
506
CMTS receives request for transmission opportunity from CM, which includes the CM ID
508
CMTS allocates a transmission opportunity to CM in response to CM request
510
CMTS uses CM ID from the request for transmission opportunity to access protocol indicator associated with CM ID
512
CMTS processes data transmitted by CM during allocated transmission opportunity in accordance with the data transfer protocol indicated by the indicator
514

FIG. 5

TOTTOT" TBZE2660

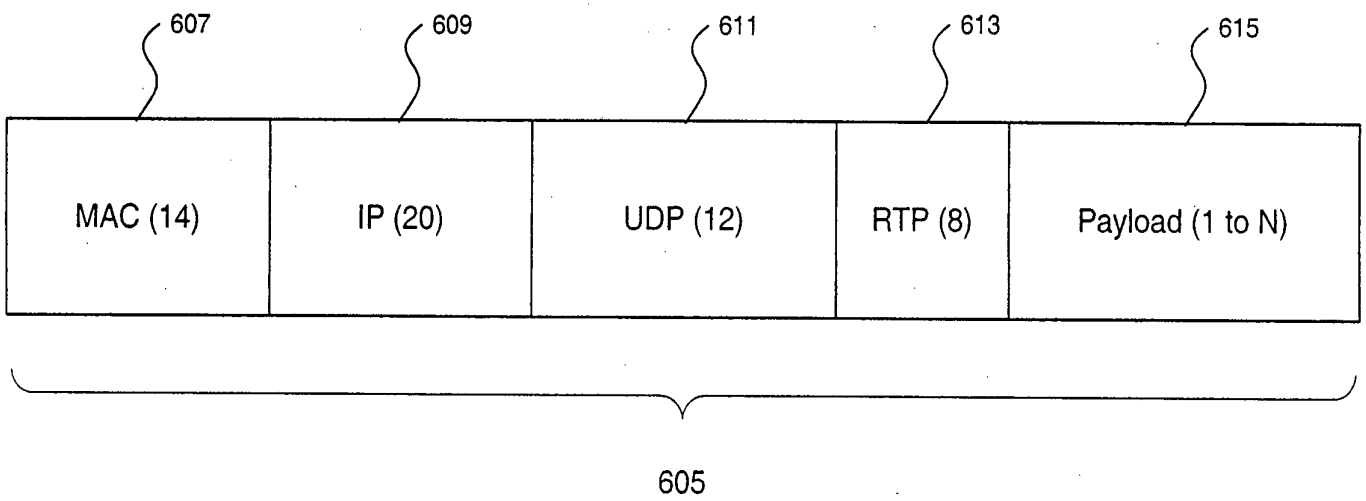


FIG. 6A

TOTTOT" TBZEZ660

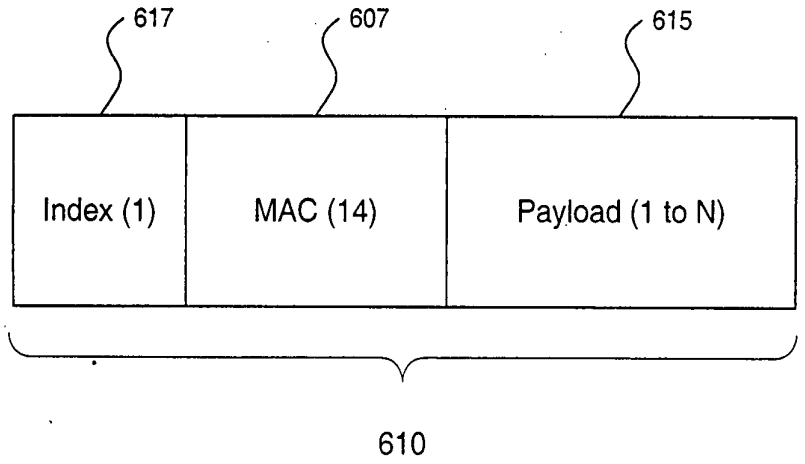


FIG. 6B

EUFIOF" FBZE2660

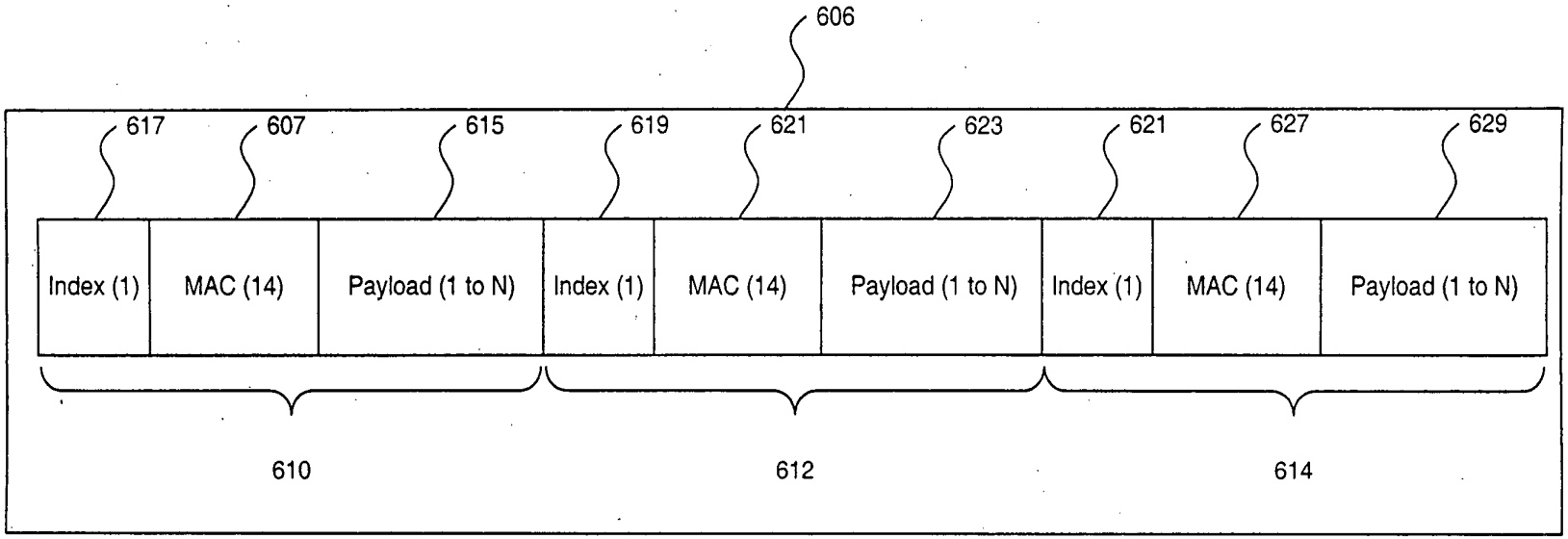


FIG. 6C

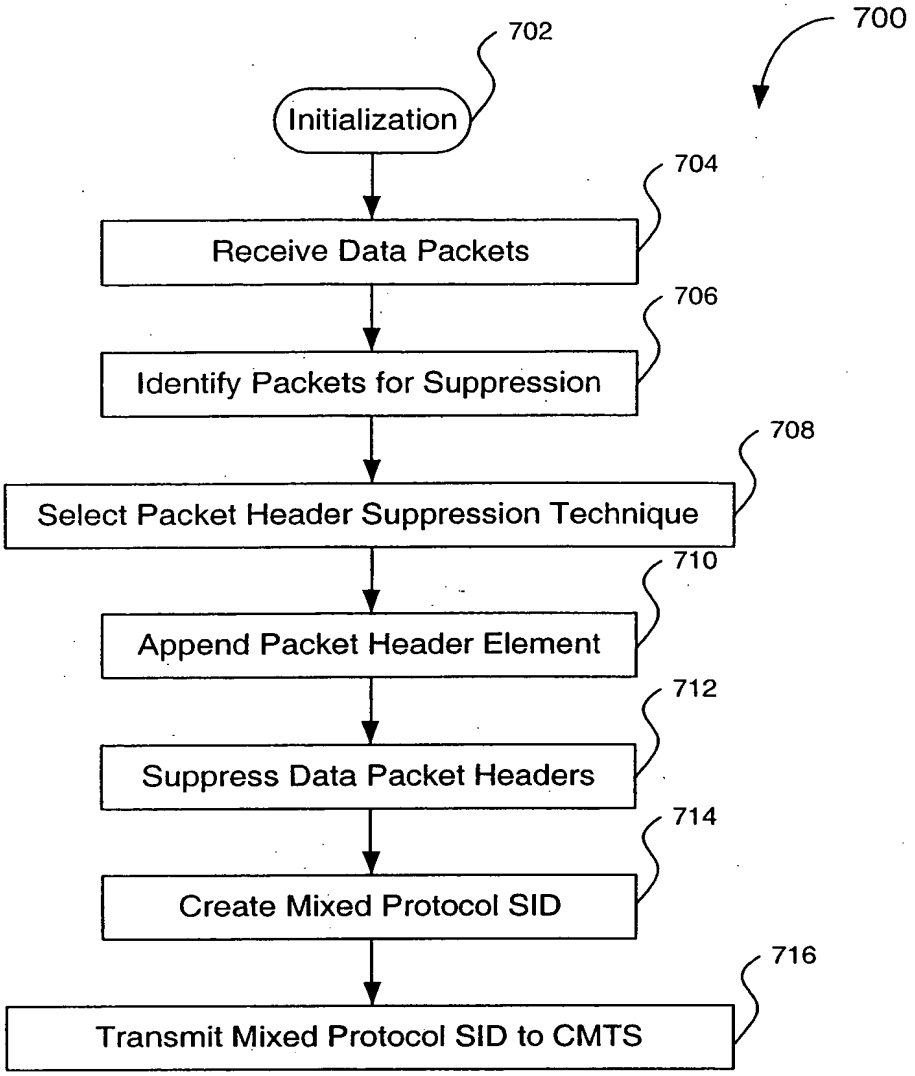


FIG. 7

0102-40.vsd/4

EDITED: 1849600

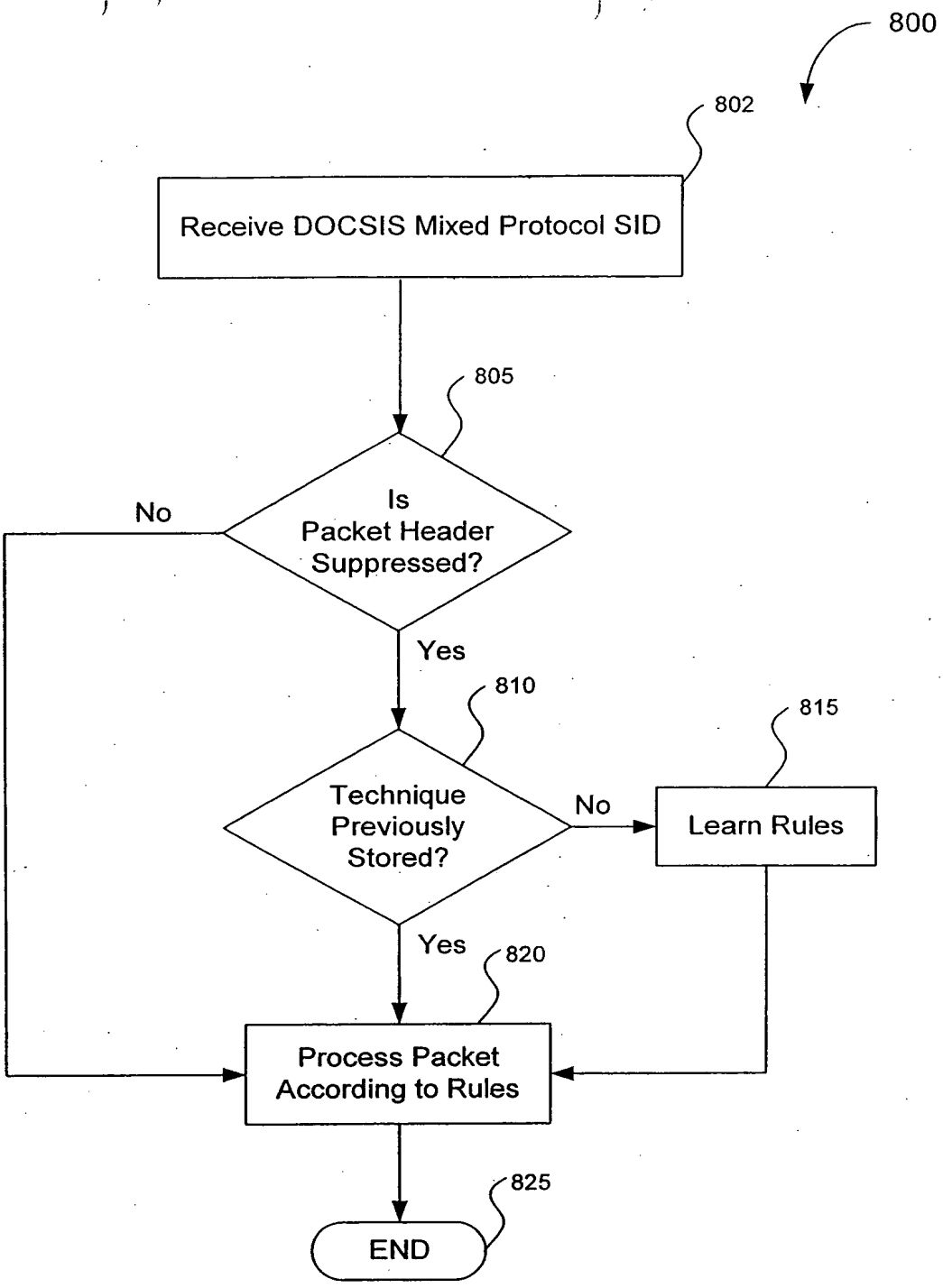


FIG. 8

FOTIOT" TBZEZ660

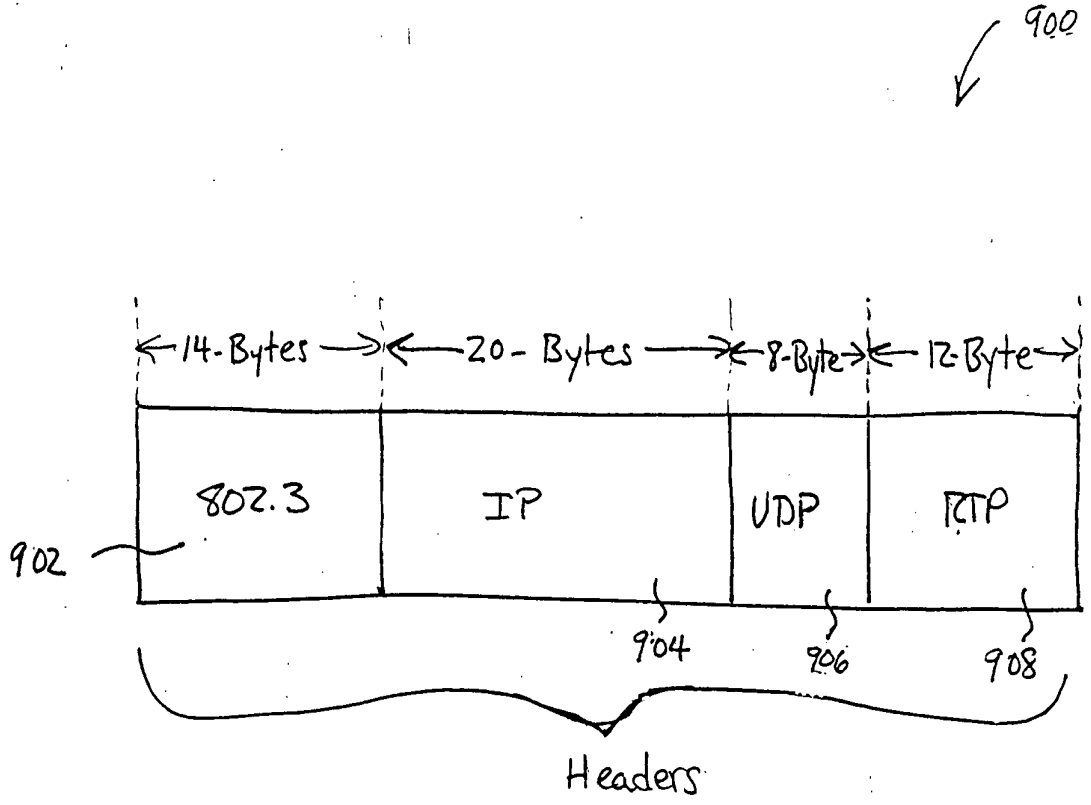


FIG. 9A

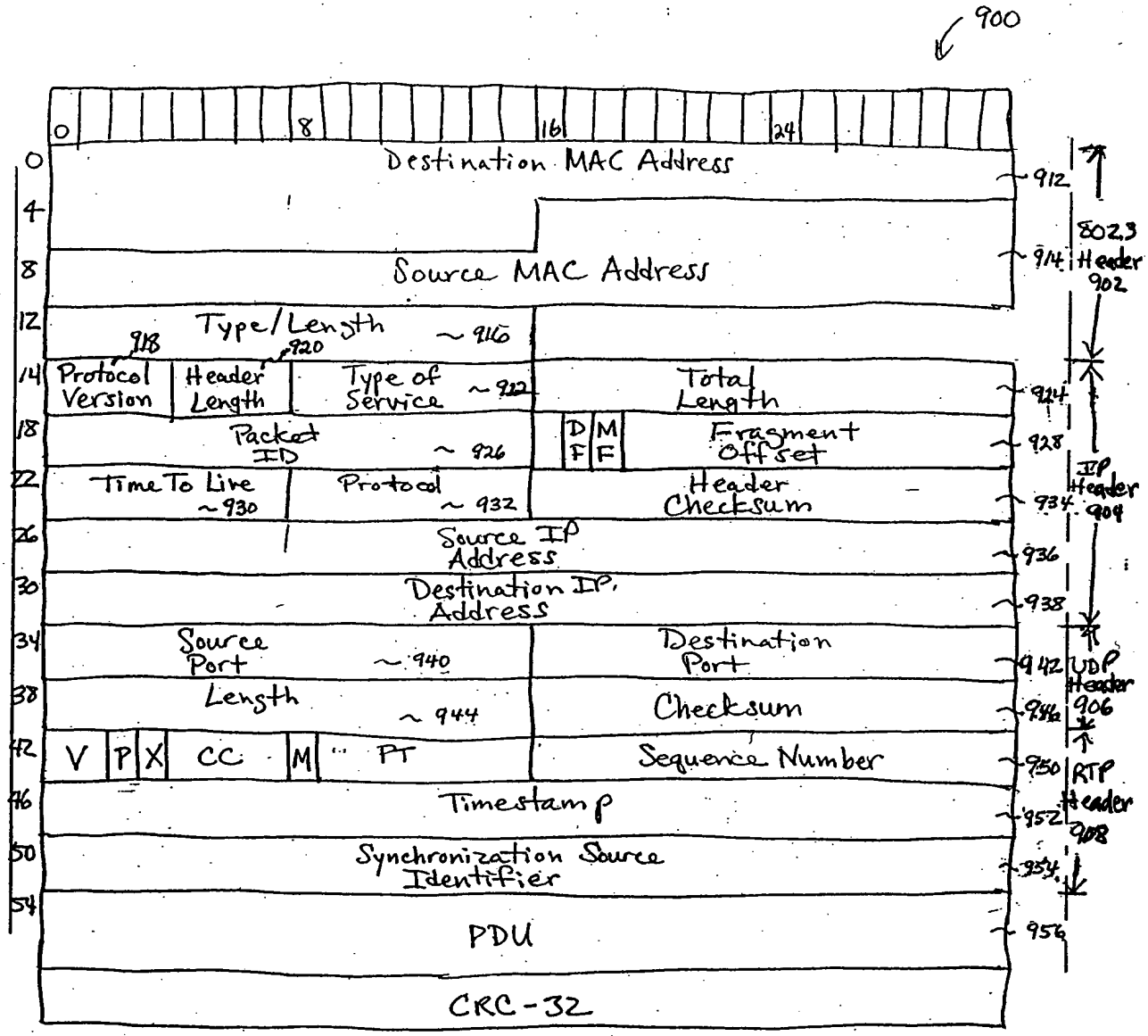

900
0
8
16
24
0
Destination MAC Address
912
4
8
Source MAC Address
914
802.3 Header 902
12
Type/Length
916
14
Protocol Version
918
Header Length
920
Type of Service
922
Total Length
924
18
Packet ID
926
DF
MF
Fragment Offset
928
22
Time To Live
930
Protocol
932
Header Checksum
934
IP Header 904
26
Source IP Address
936
30
Destination IP Address
938
34
Source Port
940
Destination Port
942
38
Length
944
Checksum
946
UDP Header 906
42
V
P
X
CC
M
PT
Sequence Number
950
RTP Header 908
46
Timestamp
952
50
Synchronization Source Identifier
954
54
PDU
956
CRC-32

FIG. 9B

TOTTOT" T8Z6660

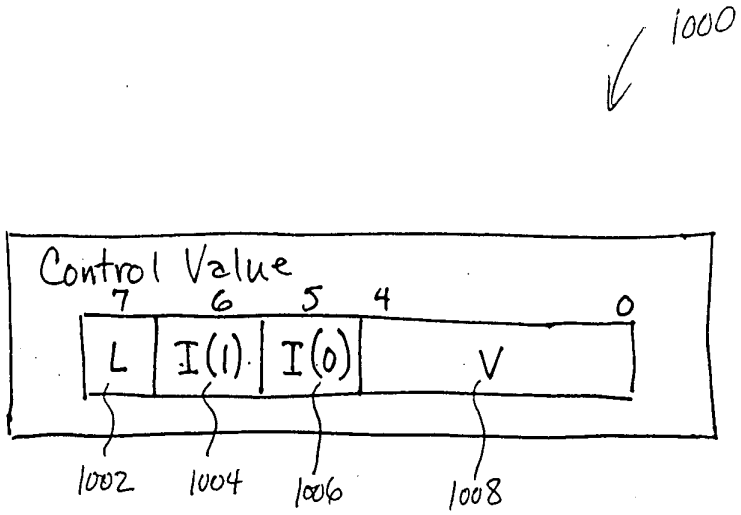


FIG. 10

FOREIGN PATENT

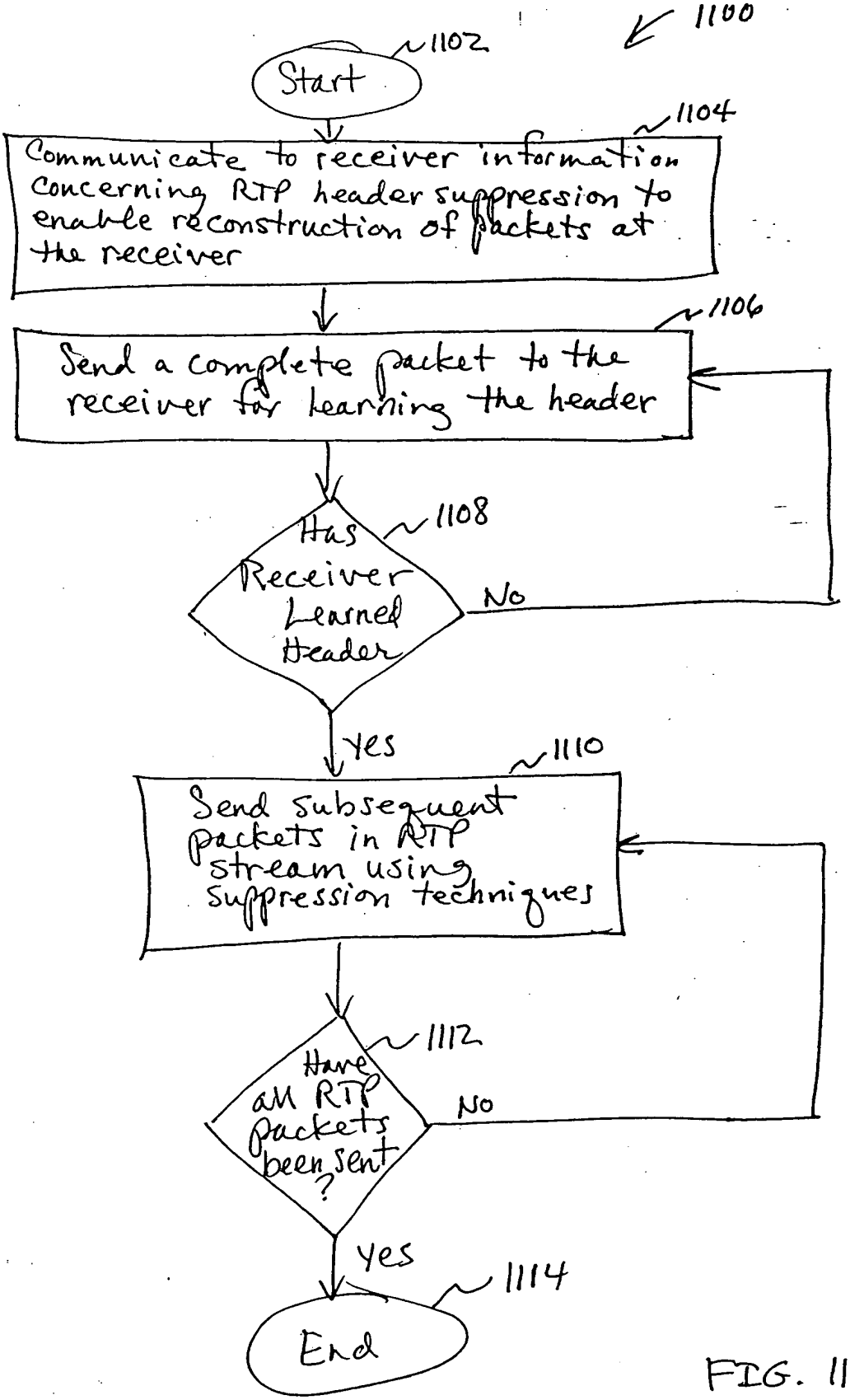


FIG. 11

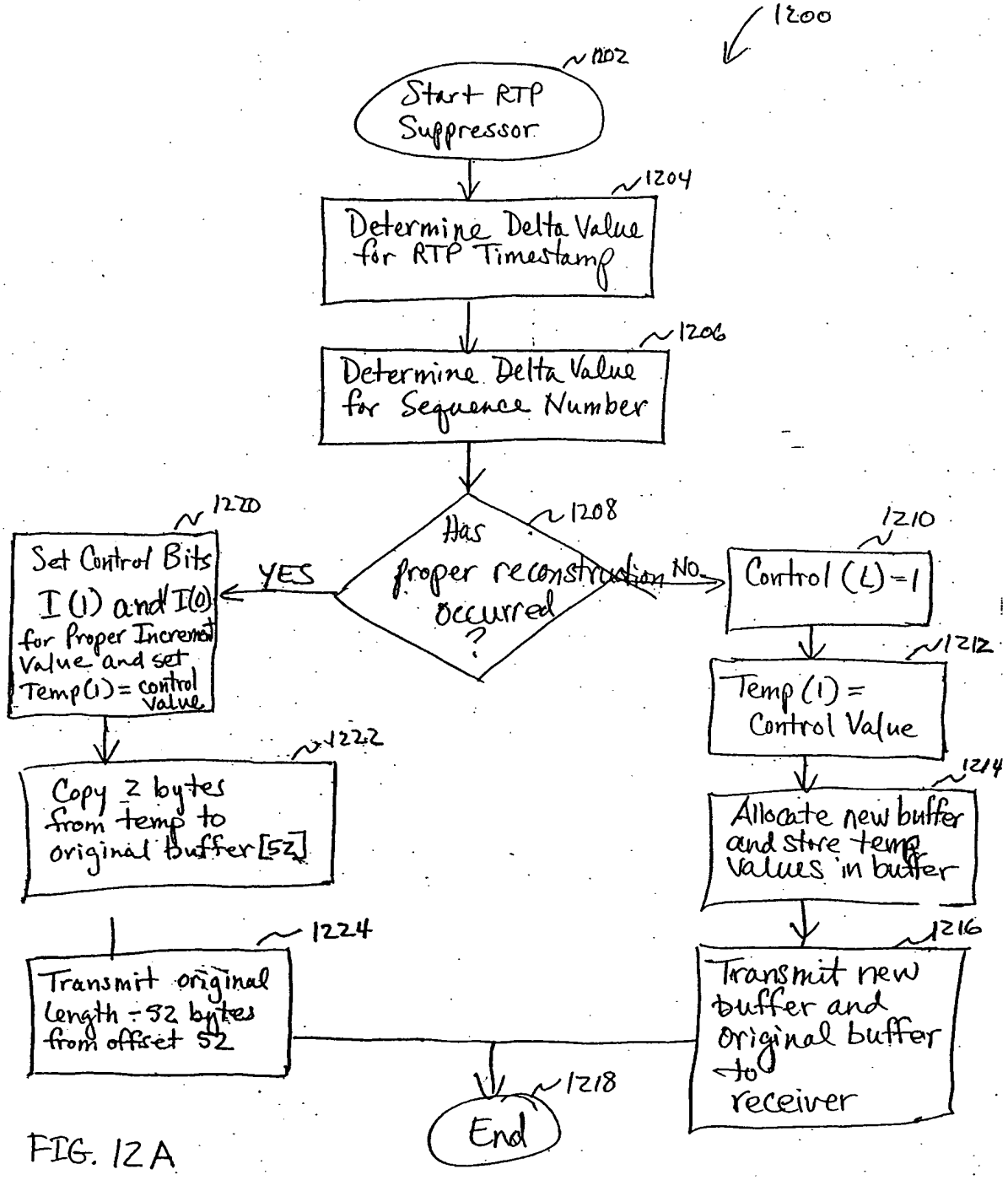

1200
1202
Start RTP Suppressor
1204
Determine Delta Value for RTP Timestamp
1206
Determine Delta Value for Sequence Number
1208
Has proper reconstruction occurred ?
YES
NO
1220
Set Control Bits I(1) and I(0) for Proper Increment Value and set Temp(1) = control value
1210
Control (L) = 1
1212
Temp (1) = Control Value
1214
Allocate new buffer and store temp values in buffer
1216
Transmit new buffer and original buffer to receiver
1222
Copy 2 bytes from temp to original buffer[52]
1224
Transmit original length - 52 bytes from offset 52
1218
End

FIG. 12A

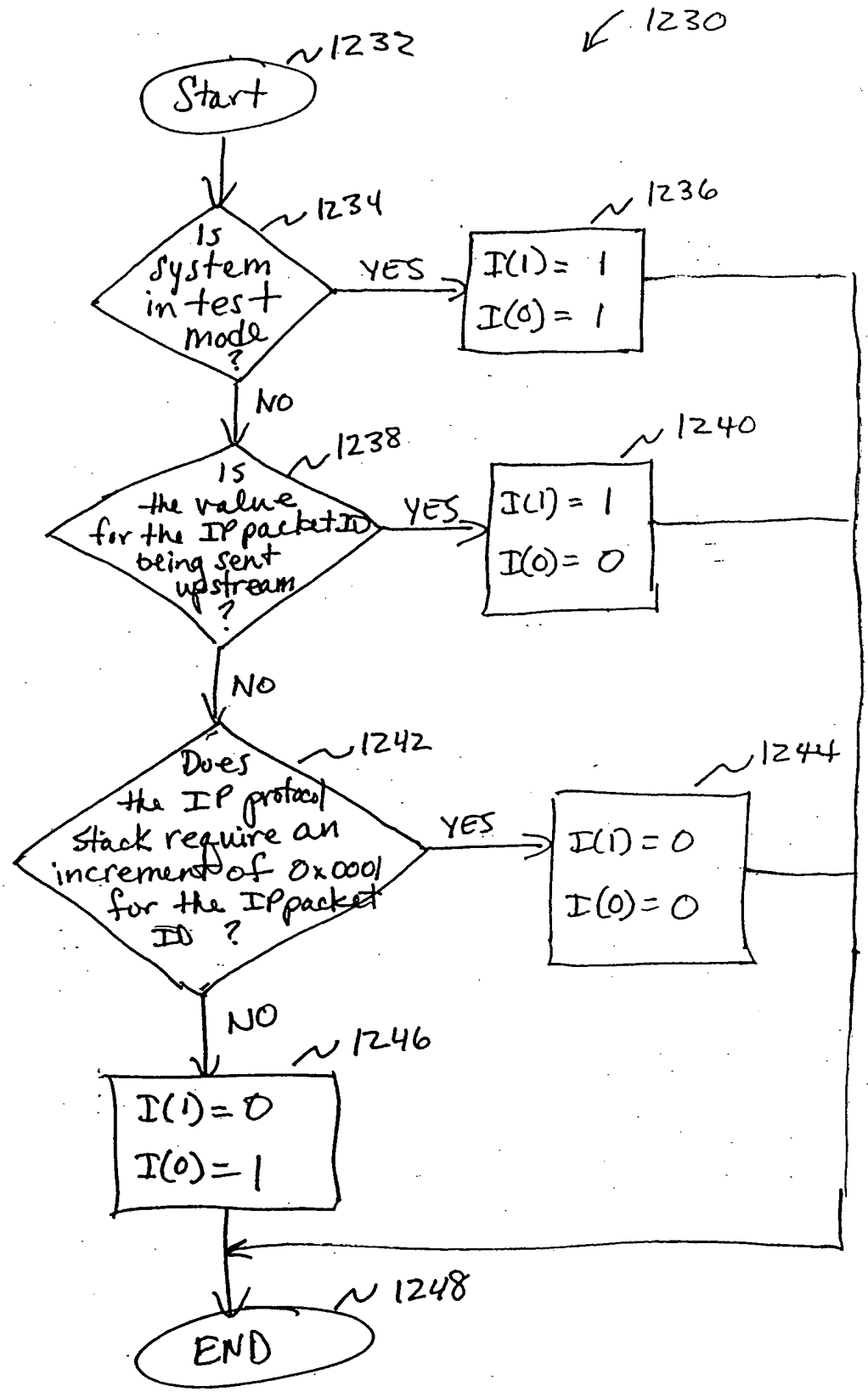

1230
1232
Start
1234
Is system in test mode ?
YES
1236
I(1) = 1
I(0) = 1
NO
1238
Is the value for the IP packet ID being sent upstream ?
YES
1240
I(1) = 1
I(0) = 0
NO
1242
Does the IP protocol stack require an increment of 0x0001 for the IP packet ID ?
YES
1244
I(1) = 0
I(0) = 0
NO
1246
I(1) = 0
I(0) = 1
1248
END

FIG. 12B

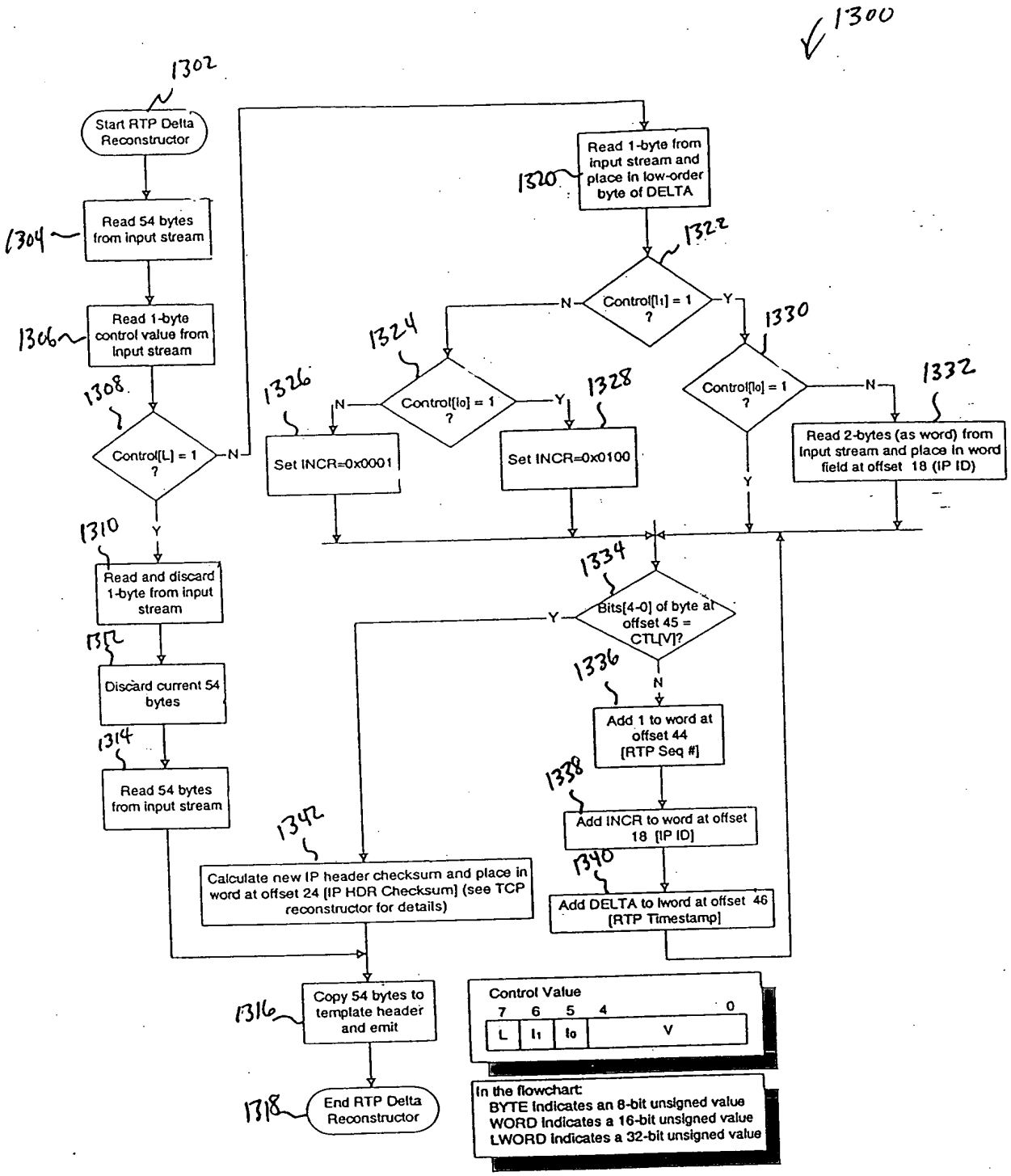

1300
1302
Start RTP Delta Reconstructor
1304
Read 54 bytes from input stream
1306
Read 1-byte control value from input stream
1308
Control[L] = 1 ?
N
Y
1310
Read and discard 1-byte from input stream
1312
Discard current 54 bytes
1314
Read 54 bytes from input stream
1316
Copy 54 bytes to template header and emit
1318
End RTP Delta Reconstructor
1320
Read 1-byte from input stream and place in low-order byte of DELTA
1322
Control[I1] = 1 ?
1324
Control[Io] = 1 ?
1326
Set INCR=0x0001
1328
Set INCR=0x0100
1330
Control[Io] = 1 ?
1332
Read 2-bytes (as word) from Input stream and place in word field at offset 18 (IP ID)
1334
Bits[4-0] of byte at offset 45 = CTL[V]?
1336
Add 1 to word at offset 44 [RTP Seq #]
1338
Add INCR to word at offset 18 [IP ID]
1340
Add DELTA to lword at offset 46 [RTP Timestamp]
1342
Calculate new IP header checksum and place in word at offset 24 [IP HDR Checksum] (see TCP reconstructor for details)
Control Value
7 6 5 4 0
L I1 Io V
In the flowchart:
BYTE indicates an 8-bit unsigned value
WORD indicates a 16-bit unsigned value
LWORD indicates a 32-bit unsigned value

FIG. 13

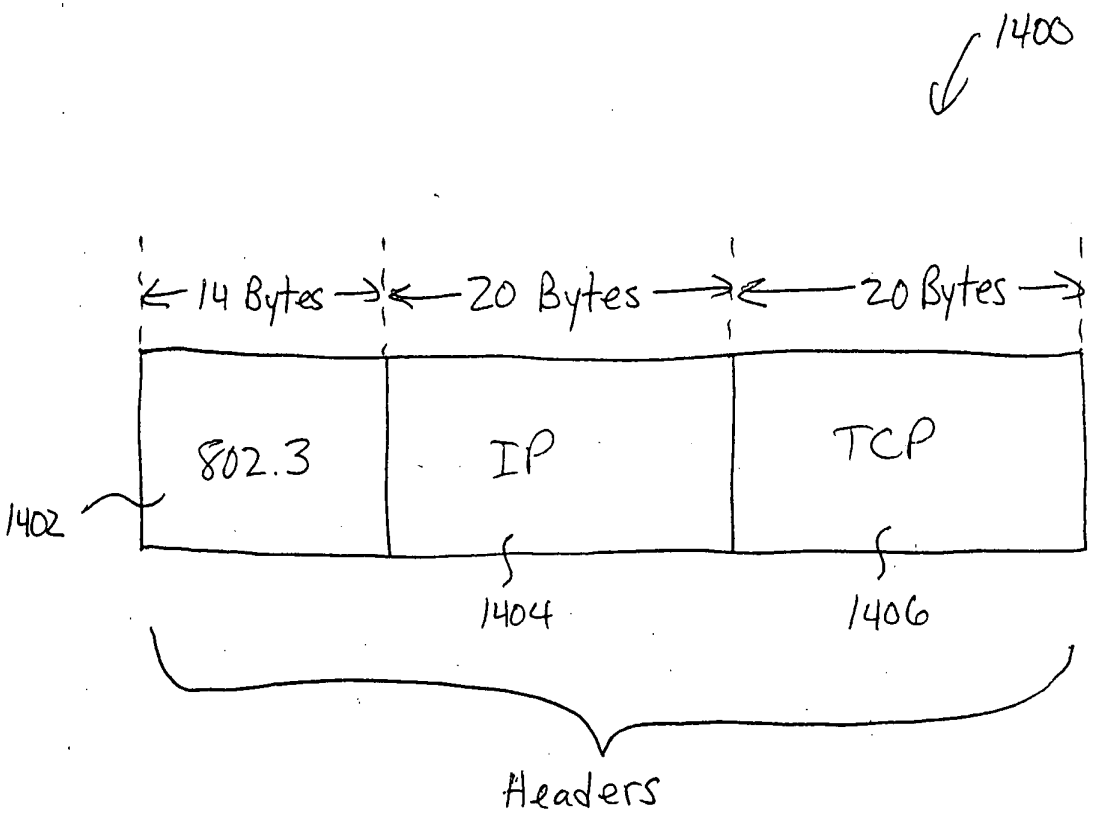

1400
14 Bytes
20 Bytes
20 Bytes
802.3
IP
TCP
1402
1404
1406
Headers

FIG. 14A

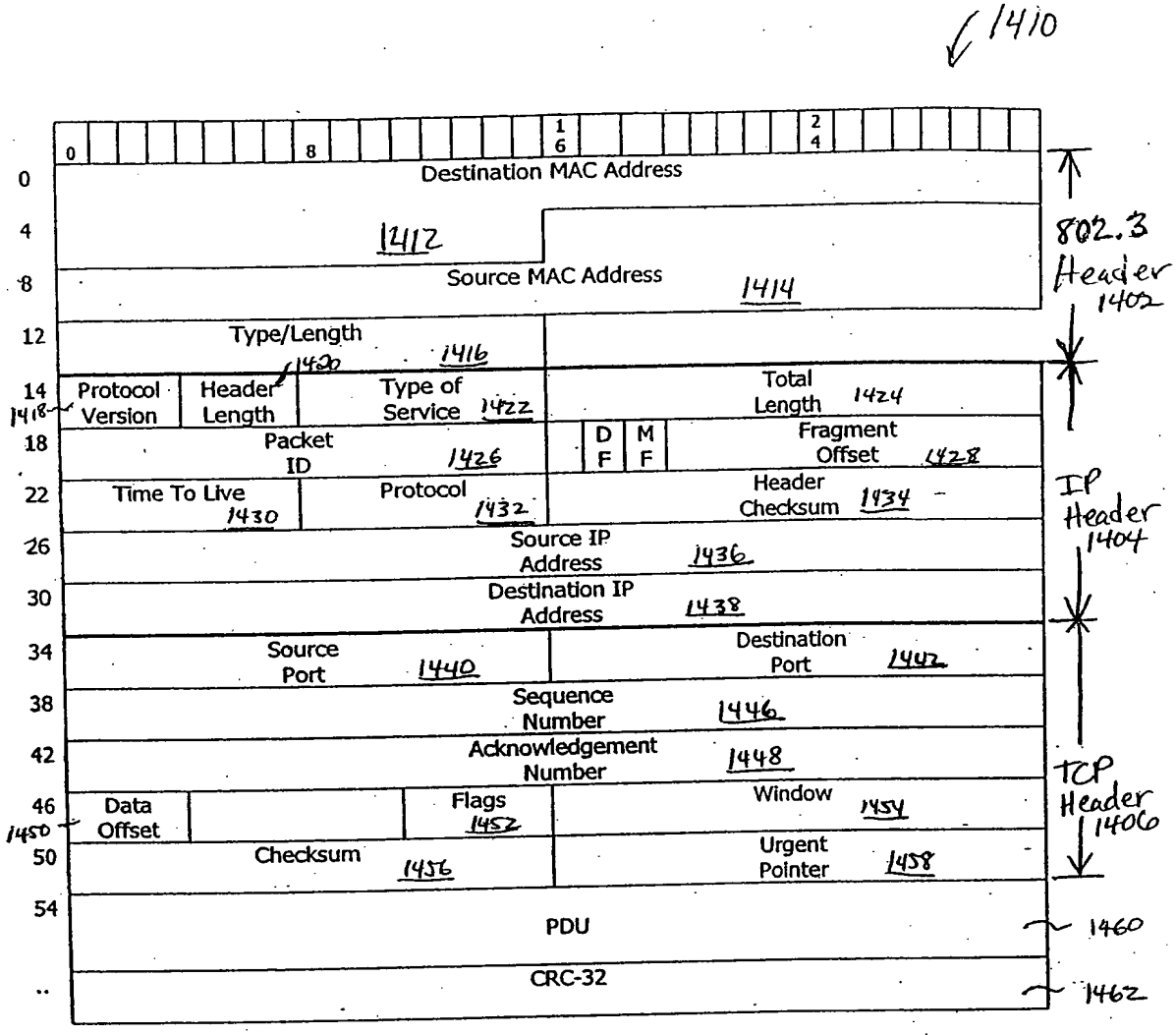

1410
0
8
16
24
0
Destination MAC Address
4
1412
8
Source MAC Address
1414
802.3 Header 1402
12
Type/Length
1416
1420
14
1418
Protocol Version
Header Length
Type of Service 1422
Total Length 1424
18
Packet ID 1426
DF
MF
Fragment Offset 1428
22
Time To Live 1430
Protocol 1432
Header Checksum 1434
IP Header 1404
26
Source IP Address 1436
30
Destination IP Address 1438
34
Source Port 1440
Destination Port 1442
38
Sequence Number 1446
42
Acknowledgement Number 1448
46
1450
Data Offset
Flags 1452
Window 1454
50
Checksum 1456
Urgent Pointer 1458
TCP Header 1406
54
PDU
1460
CRC-32
1462

FIG. 14B

1500

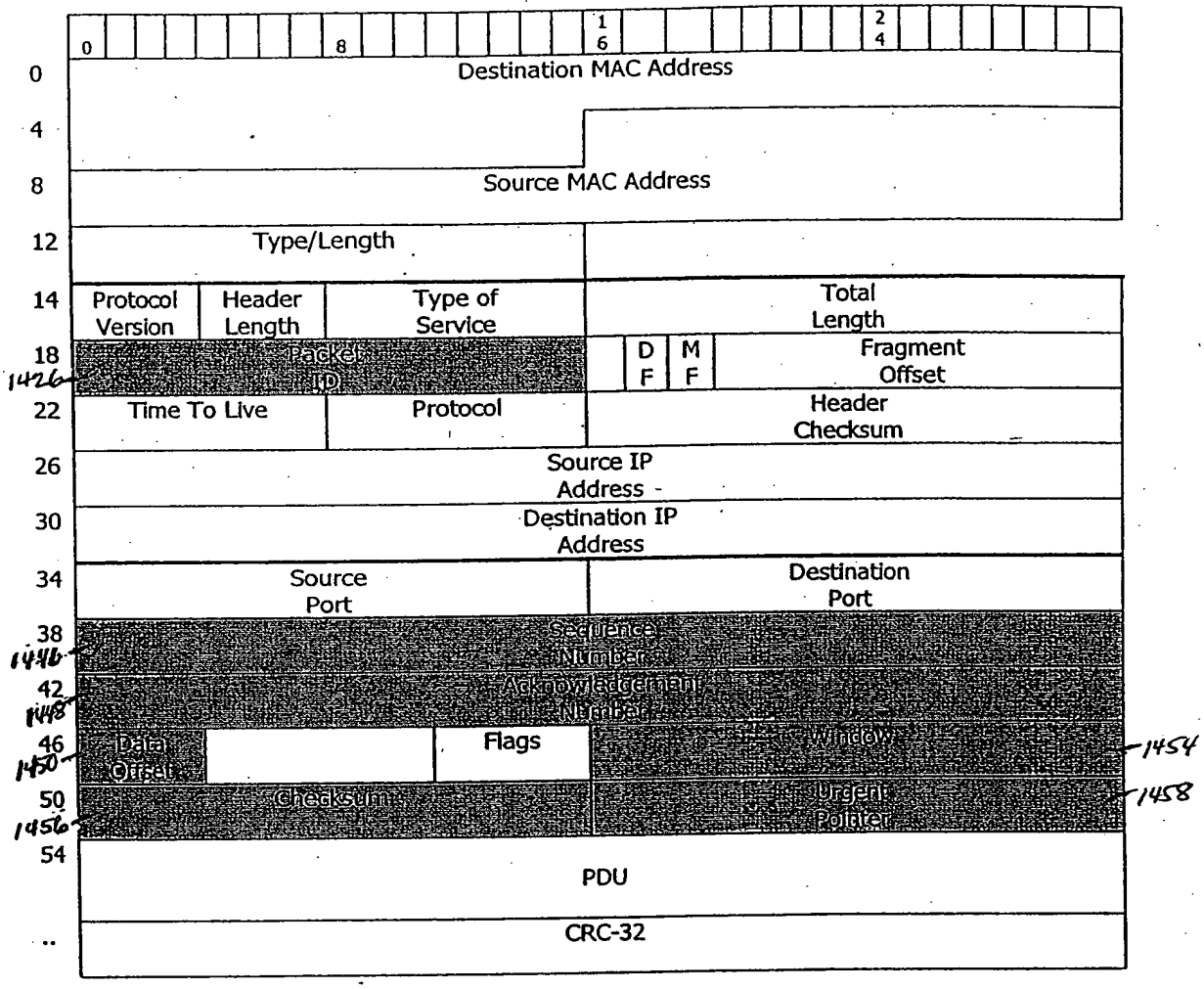

0
8
16
24
0
Destination MAC Address
4
8
Source MAC Address
12
Type/Length
14
Protocol Version
Header Length
Type of Service
Total Length
18
1426
Packet ID
D F
M F
Fragment Offset
22
Time To Live
Protocol
Header Checksum
26
Source IP Address
30
Destination IP Address
34
Source Port
Destination Port
38
1446
Sequence Number
42
1448
Acknowledgement Number
46
1450
Data Offset
Flags
Window
1454
50
1456
Checksum
Urgent Pointer
1458
54
PDU
..
CRC-32

FIG. 15

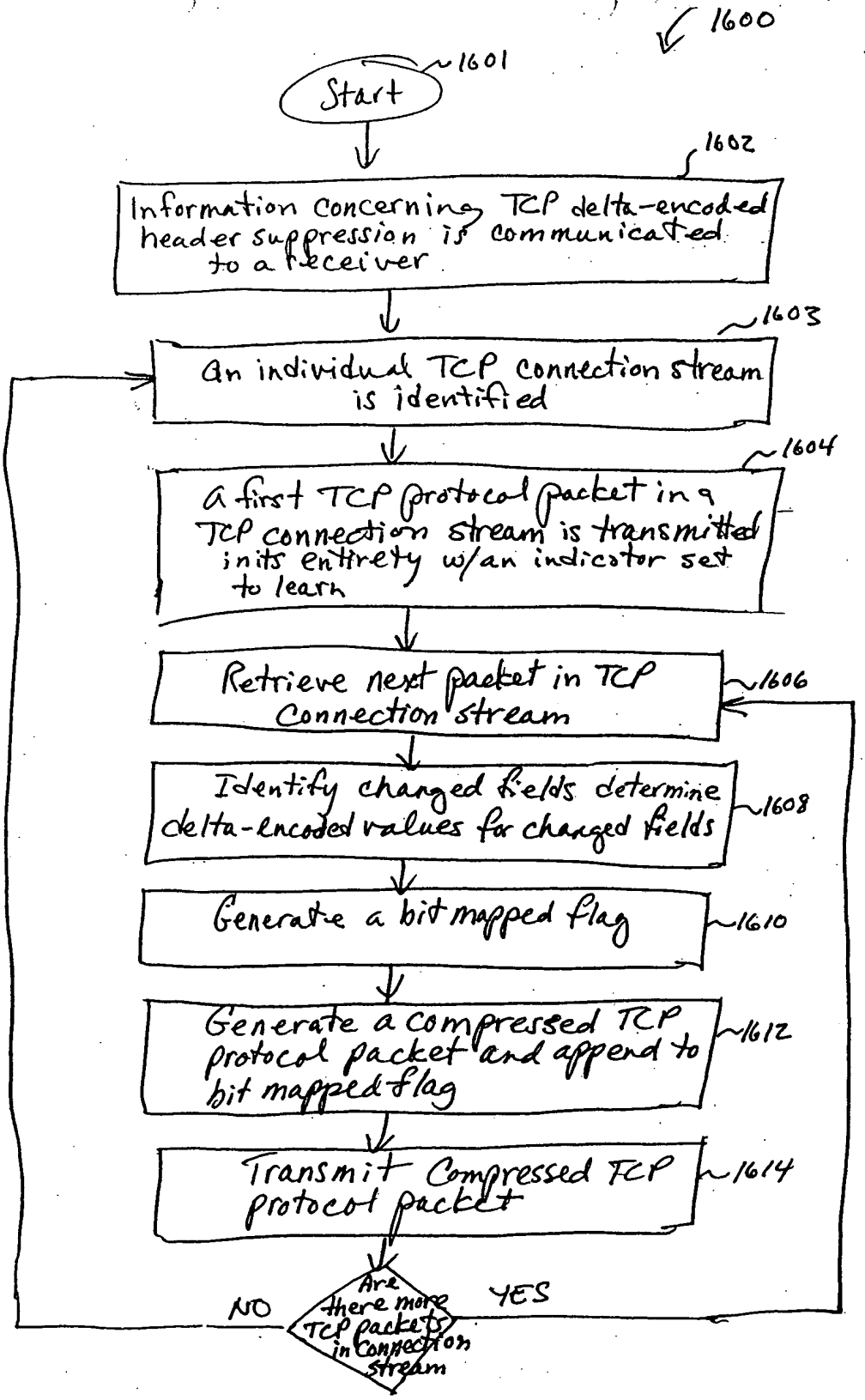

1600
1601
Start
1602
Information concerning TCP delta-encoded header suppression is communicated to a receiver
1603
An individual TCP connection stream is identified
1604
A first TCP protocol packet in a TCP connection stream is transmitted in its entirety w/an indicator set to learn
Retrieve next packet in TCP connection stream
1606
Identify changed fields determine delta-encoded values for changed fields
1608
Generate a bit mapped flag
1610
Generate a compressed TCP protocol packet and append to bit mapped flag
1612
Transmit Compressed TCP protocol packet
1614
Are there more TCP packets in connection stream
NO
YES

FIG.16A

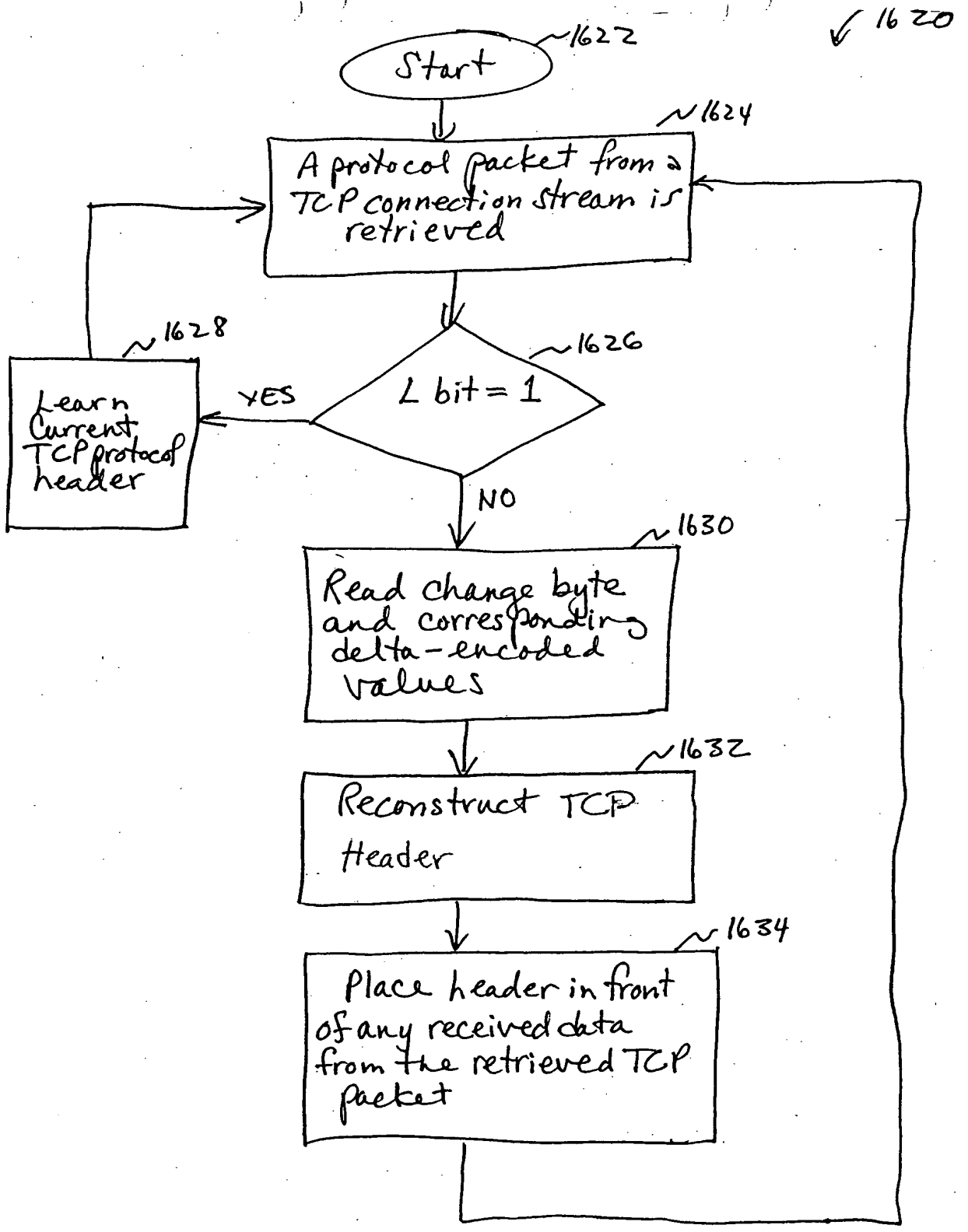

1620
Start
1622
A protocol packet from a TCP connection stream is retrieved
1624
L bit = 1
1626
YES
Learn Current TCP protocol header
1628
NO
Read change byte and corresponding delta-encoded values
1630
Reconstruct TCP Header
1632
Place header in front of any received data from the retrieved TCP packet
1634

FIG. 16B

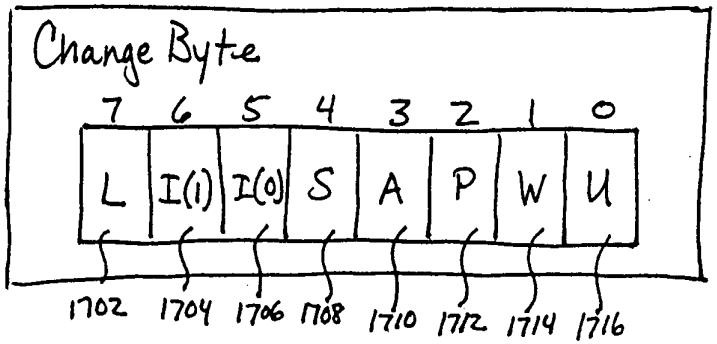

1700
Change Byte
7 6 5 4 3 2 1 0
L
I(1)
I(0)
S
A
P
W
U
1702
1704
1706
1708
1710
1712
1714
1716

FIG. 17

1800

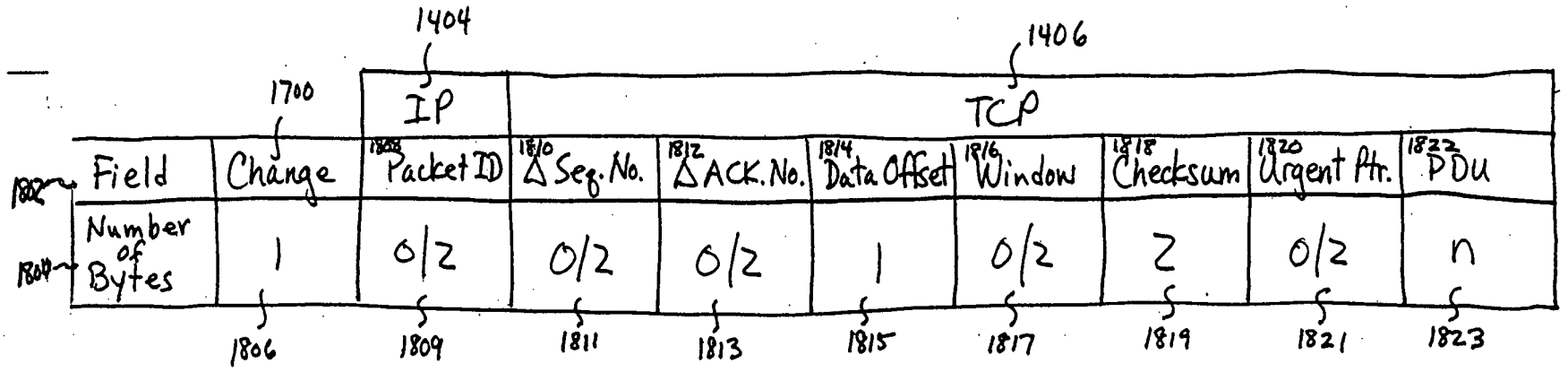

1404
1406
1700
IP
TCP
1802
Field
Change
1808 Packet ID
1810 Δ Seq. No.
1812 Δ ACK. No.
1814 Data Offset
1816 Window
1818 Checksum
1820 Urgent Ptr.
1822 PDU
1804
Number of Bytes
1
0/2
0/2
0/2
1
0/2
2
0/2
n
1806
1809
1811
1813
1815
1817
1819
1821
1823

FIG. 18

TOUCH TO SEARCH

1900

Transmit Order (Normal)

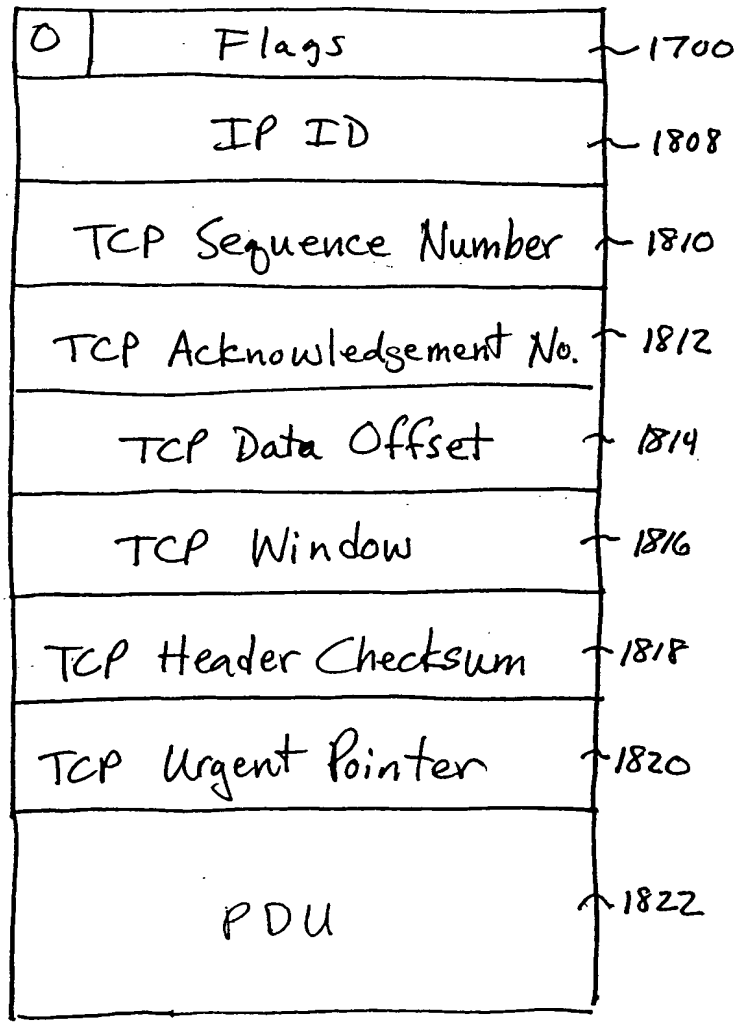


FIG. 19

EDTOT: TB/2660

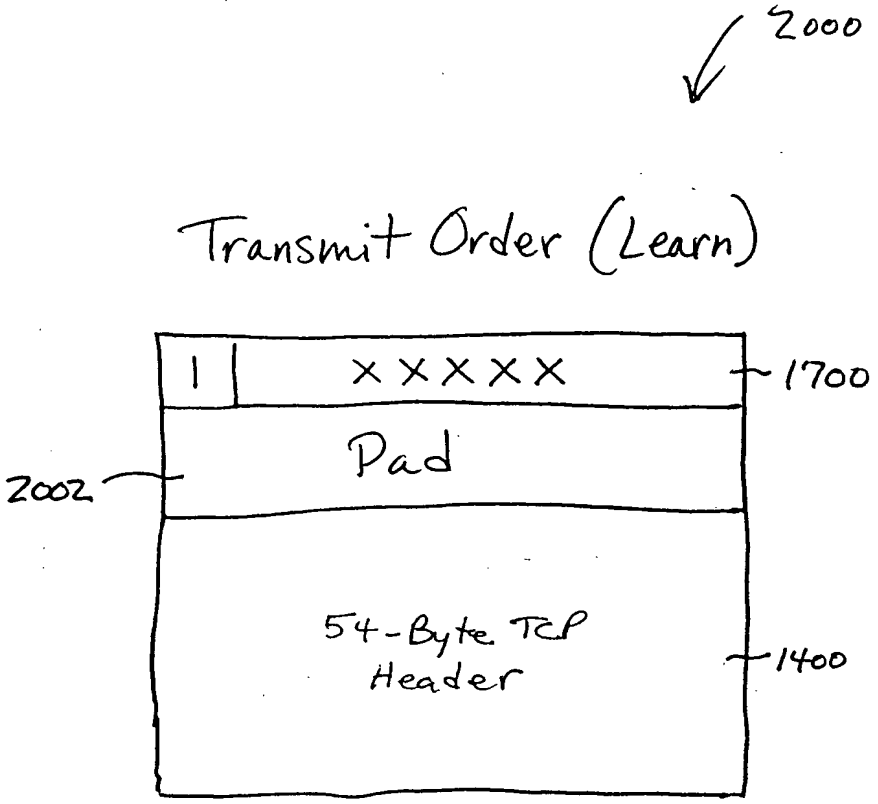


FIG. 20

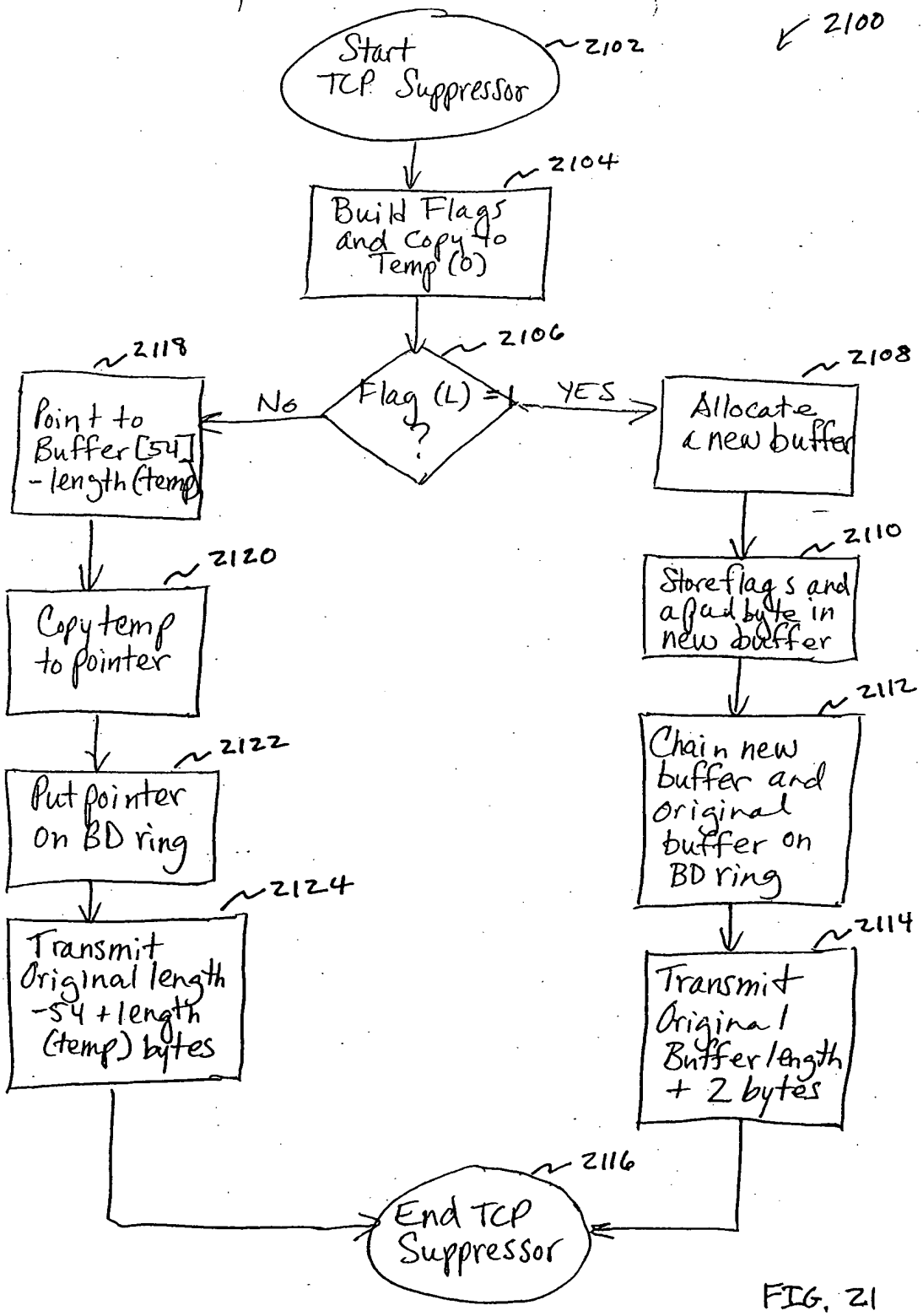

2100
Start TCP Suppressor
2102
Build Flags and Copy to Temp (0)
2104
Flag (L) = 1 ?
2106
NO
YES
Point to Buffer [54] - length (temp)
2118
Allocate a new buffer
2108
Copy temp to pointer
2120
Store flags and a pad byte in new buffer
2110
Put pointer on BD ring
2122
Chain new buffer and original buffer on BD ring
2112
Transmit Original length -54 + length (temp) bytes
2124
Transmit Original Buffer length + 2 bytes
2114
End TCP Suppressor
2116

FIG. 21

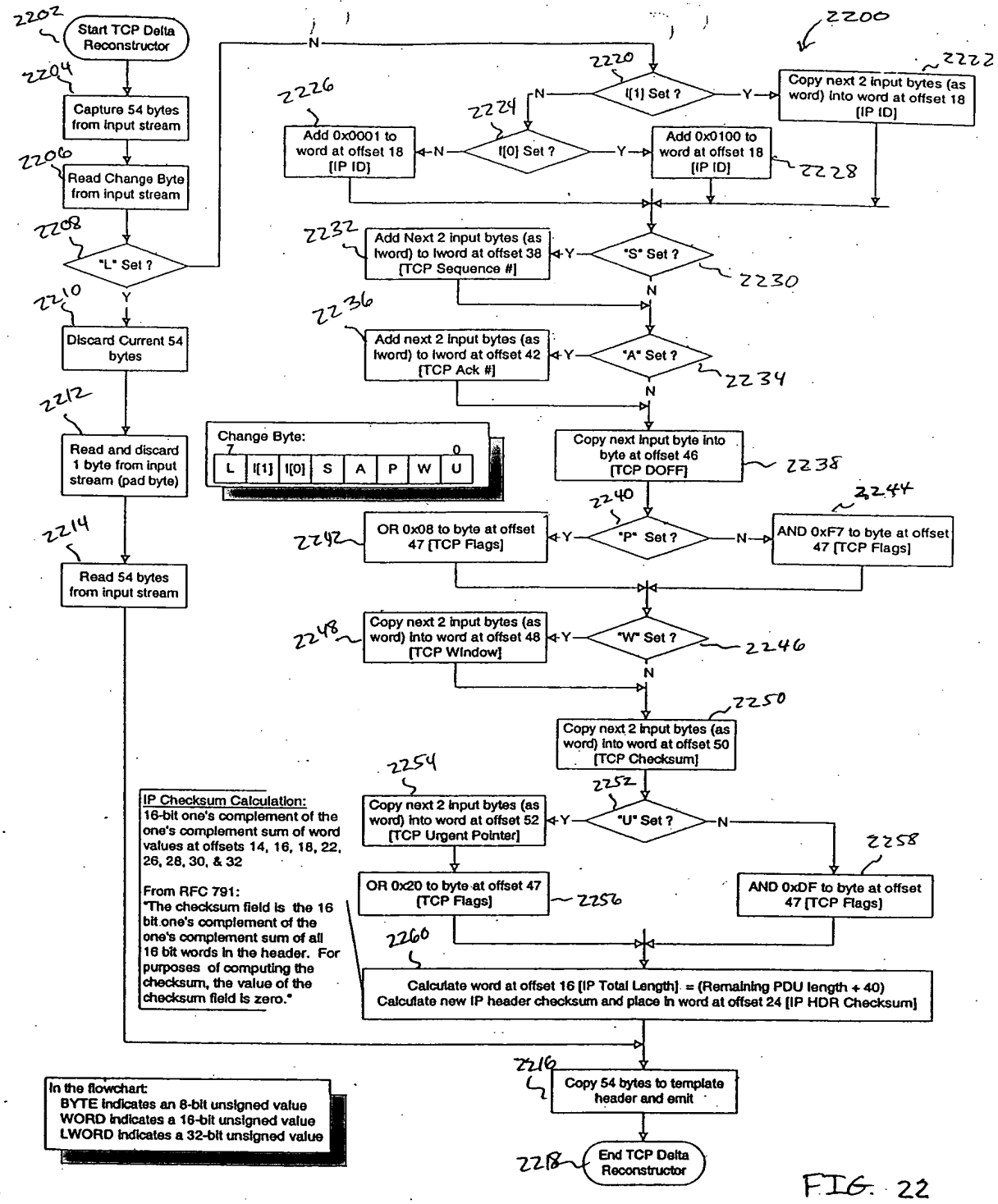

2200
2202
Start TCP Delta Reconstructor
2204
Capture 54 bytes from input stream
2206
Read Change Byte from input stream
2208
"L" Set ?
N
Y
2210
Discard Current 54 bytes
2212
Read and discard 1 byte from input stream (pad byte)
2214
Read 54 bytes from input stream
Change Byte:
7 0
L | I[1] | I[0] | S | A | P | W | U
2220
I[1] Set ?
N
Y
2222
Copy next 2 input bytes (as word) into word at offset 18 [IP ID]
2224
I[0] Set ?
N
Y
2226
Add 0x0001 to word at offset 18 [IP ID]
2228
Add 0x0100 to word at offset 18 [IP ID]
2230
"S" Set ?
Y
N
2232
Add Next 2 input bytes (as lword) to lword at offset 38 [TCP Sequence #]
2234
"A" Set ?
Y
N
2236
Add next 2 input bytes (as lword) to lword at offset 42 [TCP Ack #]
2238
Copy next input byte into byte at offset 46 [TCP DOFF]
2240
"P" Set ?
Y
N
2242
OR 0x08 to byte at offset 47 [TCP Flags]
2244
AND 0xF7 to byte at offset 47 [TCP Flags]
2246
"W" Set ?
Y
N
2248
Copy next 2 input bytes (as word) into word at offset 48 [TCP Window]
2250
Copy next 2 input bytes (as word) into word at offset 50 [TCP Checksum]
2252
"U" Set ?
Y
N
2254
Copy next 2 input bytes (as word) into word at offset 52 [TCP Urgent Pointer]
2256
OR 0x20 to byte at offset 47 [TCP Flags]
2258
AND 0xDF to byte at offset 47 [TCP Flags]
2260
Calculate word at offset 16 [IP Total Length] = (Remaining PDU length + 40)
Calculate new IP header checksum and place in word at offset 24 [IP HDR Checksum]
2216
Copy 54 bytes to template header and emit
2218
End TCP Delta Reconstructor
IP Checksum Calculation:
16-bit one's complement of the one's complement sum of word values at offsets 14, 16, 18, 22, 26, 28, 30, & 32
From RFC 791:
"The checksum field is the 16 bit one's complement of the one's complement sum of all 16 bit words in the header. For purposes of computing the checksum, the value of the checksum field is zero."
In the flowchart:
BYTE indicates an 8-bit unsigned value
WORD indicates a 16-bit unsigned value
LWORD indicates a 32-bit unsigned value

FIG. 22

TOTTOT"TBZEZ660

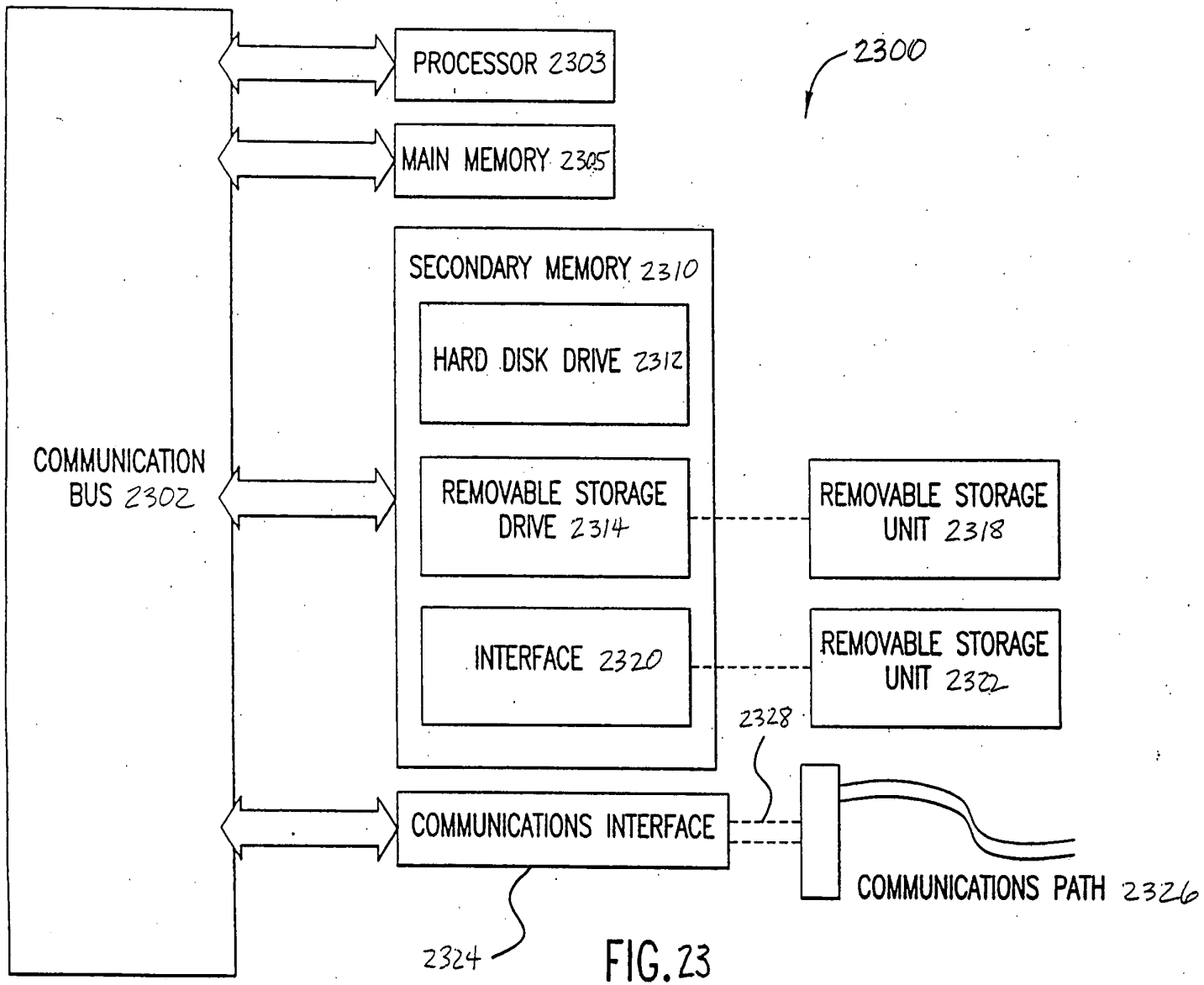


FIG. 23